浮動小数点数の「けち表現」について (計算の手間とデータ構造)

AUTHOR(S):

小木曽, 治比古

CITATION:

小木曽, 治比古. 浮動小数点数の「けち表現」について (計算の手間とデータ構造). 数理解析研究所講究録 1975, 250: 62-75

ISSUE DATE:

1975-09

URL:

http://hdl.handle.net/2433/105702

数理解析研究所講究録
250巻 1975年 62-75

62

浮動小数点数の「けち表現」について

東工大 情報科学科 小木曽治比古

§1 まえがき

最近の計算機では浮動小数点演算に関して，16進1語32ビット（16進6桁）の構造をとっているものが少なくない．しかしこれでは精度の不足が感じられることが多い．本稿では1語32ビットという条件は与えられたものとして，より良い方式がないか検討してみたい．その一つの方法として2進表示を使い，正規化された形では仮数部の最上位のビットは常に定まっていることに着目して1ビットの節約をはかる方法がある（これを以後「けち表現」と呼ぶ），これはPDP 11/45等では実際に用いられている[1]．また種々の浮動小数点表示方式の精度に関する比較論が[2]にある．

本稿では，1°この方式など8種類の方式について行なったソフトウェア実験の結果，2°「けち表現」のHP-21MX計算機でのマイクロプログラム実験の結果，及び3° 2,3の理論

的検討　を行なう．

§2　準備

以下の話をはっきりさせるために次のような形式の浮動小数点数を考える．

S	E	M

S，E，Mはビット列で，Sは符号部（長さ1ビット），Eは指数部（長さ1ビット以上），Mは仮数部（長さ0ビット以上）とする．　s，m，eを以下のように定義する．

$$s=\begin{cases}1 & S=0\ \text{のとき}\\ -1 & S=1\ \text{のとき}\end{cases}$$

m：Mを左端に小数点を置いた2進数とみたときの値．

e：左端の1ビットをゲタとみた正負の整数値と定義する．

（たとえばEが長さ8ビットのときは，Eを2進数とみた値から2^7を引いたものとする．　eの選び方は他にも種々あり得るが簡単のためこのように定めておく）

実数を浮動小数点数で近似するとき，本稿では特に断わらない限り「丸め」を使うものとする．

次の8種類の方式を考える．　ただしfは，S，E，Mの三組によって表わされる浮動小数点数の値とする．

a. log 方式（以後Sl方式と呼ぶ）

$$f = \Delta \times 2^{\frac{e}{K}} \quad (K\text{は定数}) \qquad \cdots(注)$$

b. 2進「けち」方式（以後Sg2方式と呼ぶ）

$$f = \Delta \times (m+1) \times 2^e \qquad \cdots(注)$$

c. 2進方式（以後Sn2方式と呼ぶ）

$$f = \Delta \times m \times 2^e$$

d. 4進「けち」方式（以後Sg4方式と呼ぶ）

$$f = \Delta \times (m+\tfrac{1}{3}) \times 4^e \qquad \cdots(注)$$

e. 4進方式（以後Sn4方式と呼ぶ）

$$f = \Delta \times m \times 4^e$$

f. 16進「けち」方式（以後Sg16と呼ぶ）

$$f = \Delta \times (m+\tfrac{1}{15}) \times 16^e \qquad \cdots(注)$$

g. 16進方式（以後Sn16と呼ぶ）

$$f = \Delta \times m \times 16^e$$

h. 16進切り捨て方式（以後St16と呼ぶ）

$f = \Delta \times m \times 16^e$ 但し切り捨てを行なう。

(注) 全部のビットが0のときは $f=0$ とする。

§3 ソフトウェア実験

§2で示した8種類の浮動小数点数表示方式の性能をソフトウェアのシミュレーションによって調べたので，本節ではこれについて述べる．以下の実験では指数部Eのビット数を2進法(S_{g2}, S_{n2})では9，4進法(S_{g4}, S_{n4})では8，16進法($S_{g16}, S_{n16}, S_{t16}$)では7とし，log方式($S_\ell$)のkを$2^{22}$と定めた．これは表わせる数の範囲を同じにするためである．
誤差の評価には相対誤差を2^{23}倍したもの(以後E_cと呼ぶ)をおもに使う．使用した計算機はFACOM 230-45Sである．
実験に使った疑似乱数の発生方法は次のとおりである．まず(-1, 1)の倍精度一様乱数を作る．これは参考文献[3]においてAlgorithm Mと呼ばれている方法を使って発生した．この(-1, 1)一様乱数を使って倍精度の$(-2^{16}, -2^{-16})$及び$(2^{-16}, 2^{16})$の範囲で分布するlog一様乱数(logをとったものが一様分布するような乱数)を作った．以後このlog一様乱数を単に乱数と呼ぶ．

3.1 素演算

まず確認のため2000個の乱数を発生させ，それを各方式に変換したときに出る誤差を調べた．次には1000組(1組2個)の乱数に倍精度の加算，乗算，除算を行なったとき(

これを真の値とみなす）と，この演算を各方式に変換して行なったときとの差を調べた．加算では相対誤差のかわりに誤差を2つの数の絶対値の和で割ったものを使った．以上の実験からEcの平方の平均の平方根（以下Ermsと呼ぶ）を求め，結果をT3.1に示す．

	変換	加算	乗算	除算
S_l	0.404	0.534	0.558	0.584
S_{g2}	0.420	0.559	0.760	0.705
S_{n2}	0.828	1.138	1.433	1.462
S_{g4}	0.523	0.709	0.938	0.890
S_{n4}	0.649	0.878	1.073	1.121
S_{g16}	0.902	1.120	1.539	1.570
S_{n16}	0.954	1.161	1.623	1.647
S_{t16}	1.824	3.096	4.521	2.673

T3.1

この結果から，1)精度は$S_l, S_{g2}, S_{g4}, S_{n4}, S_{n2}, S_{g16}, S_{n16}, S_{t16}$の順になっている．2)$S_{t16}$の誤差には偏りがあるため乗算と除算でのErmsの差が大きい．ということがわかる．

3.2 平方根と立方根の計算

Newton法によって乱数xの平方根と立方根を各形式で求めそれの誤差を調べた．近似式は下記のとおりで，答より大きな値，max(1,x)から始め，$a_{n+1} \geq a_n$になるまで繰り返した．

$$a_{n+1} = \frac{1}{2}\left(a_n + \frac{x}{a_n}\right) \cdots \sqrt{x}, \quad a_{n+1} = \frac{1}{3}\left(2a_n + \frac{x}{a_n^2}\right) \cdots \sqrt[3]{x}$$

これを1000個の乱数について行ない，Ermsを求めた．この

結果をT3.2に示す．

S_{g4}が平方根では割に精度が悪く，立方根では良いのは，1/2は誤差なく表わせないが1/3は完全に表わせるからだと思われる．

	平方根	立方根	E(Ec)	ErmS
S_{ℓ}	0.519	0.613	1.757	3.589
S_{g2}	0.601	0.665	2.102	3.969
S_{n2}	1.219	1.950	4.733	10.325
S_{g4}	1.167	0.813	2.733	4.926
S_{n4}	1.066	2.040	3.095	6.168
S_{g16}	1.599	1.489	4.023	7.657
S_{n16}	1.568	1.537	4.364	8.227
S_{t16}	3.077	5.058	10.820	20.385
	T3.2		T3.3	

3.3 逆行列の計算

乱数から[-1, 1]に入る数だけ取り出して5×5の行列を500個作り，これの逆行列をGauss-Jordan法によって求めた．誤差を評価するために，Ecとして

$$\frac{|Md-Ms|}{|Md|}\times 2^{23}$$

を使った．(Mdは倍精度で，Msは各方式で求めた逆行列でノルムとしては全部の要素の絶対値の和を採った) この実験によってEcの平均値とErmsを求め，その結果をT3.3に示す．

Ermsでは大きな値を取るデータの影響が強く現われて良い結果が得られなかった．S_{n2}よりもS_{n16}の方が精度が良か

った．これは計算の途中に現われる数の分布がlog一様分布より一様分布に近くなることを示していると思われる．

§4 マイクロプログラム実験

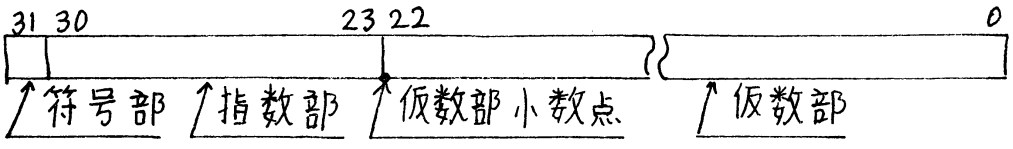
次に本節では，前節から精度が良くて実用的なことがわかったSg2方式をHP 21MXのマイクロプログラミングで実験してみたのでその結果をシステム組込みのSn2方式と比較しながら示す．

◦ハードウェアシステムの概要

マイクロプログラムは1語24ビットの垂直形で2バス方式（T←A+Bは，Reg←A; T←Reg+Bとしなければならない）であり，最大語数は4K語であるが，その内の1K語をシステムが基本命令等のために使っている．1マイクロ命令は325nsで実行される．実験に使用したHP 21MXにはWCS (writable control store) が256語ついている．

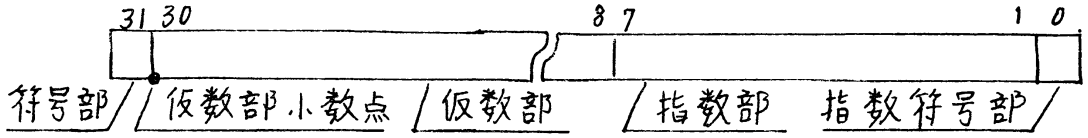
◦浮動小数点数の形

本実験の形式（Sg2方式）



指数は2^7-1のゲタはかせである．仮数の値は仮数部を2進小数とみた値に1足した値である．

HP 21MX の標準形式（Sn_2方式）



指数部，仮数部とも２の補数表示である．

◦結果と検討

速度と大きさの結果をT4.1に示す．本文の形式が劣るようにみえるがこれは，たまたまえらんだ指数部の置き方による．指数部を右へもって行けば両者はほとんど同じとなることが，わかった．すなわちSg_2方式では浮動小数点数を指数と仮数に分解する際に仮数に１加えるために３マイクロ命令余分に要するが，合成するときには正規化のためのシフトが１ビットにつき２マイクロ命令だけ省けるからである．

	Sg_2	Sn_2
語数	$191\ (277)_8$	$230\ (346)_8$
加算	35～66μs	22～54μs
減算	38～68μs	23～57μs
乗算	58～68μs	48～57μs
除算	66～74μs	41～76μs

T4.1

§5 理論的考察

5.1 方式と精度

正規化された浮動小数点数が表わす数の集合をFとする．

また $\{|f| \mid f \in F, f \neq 0\}$ の最大値と最小値を f_{max}, f_{min} とする．無限集合R（実数）の元 x（$f_{min} \leq |x| \leq f_{max}$）は有限集合Fの元 $fl(x)$ で近似されるものとする．

Fを定めている方式の指数部のビット数を2進方式に換算した値Leは

$$L_e = \log_2 \left\{ \log_2 \left(\frac{f_{max}}{f_{min}} \right) \right\}$$

である．これから標準相対誤差として

$$\left(\frac{x - fl(x)}{x} \right) \times 2^{(32-L_e)} \qquad E5.1.1$$

を考える．これが前に使ったEcである．

b 進法（普通 $b = 2^k$ である）の浮動小数点数は一般に

$s \times m \times b^e$　（s, m, e は符号，仮数，指数）

と表わせる．L_E, L_M を指数部，仮数部のビット数とすると

$$L_e = L_E + \log_2(\log_2 b) \qquad E5.1.2$$

$$0 \leq \left| \frac{x - f}{f} \right| \leq \frac{1}{2 \cdot m \cdot 2^{L_M}} \qquad (f = fl(x)) \quad E5.1.3$$

となる．E5.1.2とE5.1.3よりEcの最大値は

$$\sup_{\frac{1}{b} \leq m < 1} \frac{1}{2m \cdot 2^{L_M}} \times 2^{\{32 - L_E - \log_2(\log_2 b)\}}$$

$$= \sup_m \left\{ \frac{1}{m \log_2 b} \right\} = \frac{b}{\log_2 b} \qquad E5.1.4$$

となる．次にχがlog一様分布しているとしてErmsを求めると

$$(Erms)^2=\int_{fmin}^{fmax}(Ec)^2\frac{d\chi}{(\log_e fmax-\log_e fmin)\times\chi}$$

$$=\int_{\frac{1}{b}}^{1}\left\{\frac{1}{3(\log b)^2}\cdot\frac{1}{\chi}\right\}\cdot\frac{d\chi}{(\log b)\times\chi}$$

$$=\frac{b^2-1}{6(\log_2 b)^3\log_e 2}\qquad E5.1.5$$

となる．

S_{g2}, S_{g4}, S_{g16}の自然な拡張としてb進「けち」方式を考え，それの仮数のゲタ（普通のゲタとは逆である）をGとすると，仮数の近似する範囲（話を簡単にするため「切り捨て」を使う）は$[G,G+1)$となる．このGは条件

$$\{x\mid fmin\le|x|<fmax\}\subset\bigcup_e Ie\qquad C.5$$

を満たさなければならない．ただしIeは指数がeである浮動小数点数が近似する範囲$[-b^e\cdot(G+1),-b^e\cdot G)\cup(b^e\cdot G,b^e\cdot(G+1)]$を表わす．これがGとしてC.5を満たす最大値

$$G+1=b\cdot G$$

$$G=\frac{1}{b-1}\qquad E5.1.6$$

をとる．各けち方式ではEc, Ermsとも普通の方式の$1/(1+G)$倍である．

S_lでは$k=2^n$とすると

$$Le = \log_2\left(\frac{2^{31}}{2^m}\right) = 31 - m \qquad E5.1.7$$

$$0 \le \left|\frac{x - fl(x)}{x}\right| \le \frac{1 - 2^{\frac{1}{k}}}{2} = \frac{\log_e 2}{2k} \qquad E5.1.8$$

である．これよりEの最大値とErmsは $\log_e 2$ と $(\log_e 2)/\sqrt{3}$ である．以上の結果をT5.1に示す．

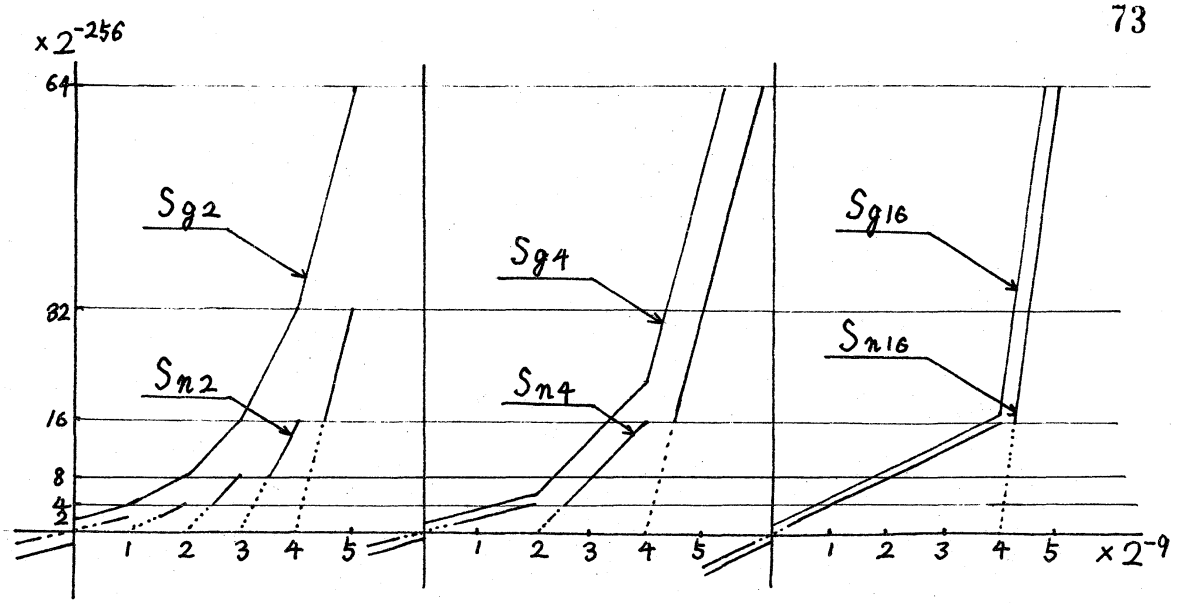
	sup(Ec)	Erms
S_l	0.693	0.400
S_{g2}	1.0	0.425
S_{n2}	2.0	0.849
S_{g4}	1.5	0.504
S_{n4}	2.0	0.671
S_{g8}	2.333	0.655
S_{n8}	2.667	0.749
S_{g16}	3.75	0.918
S_{n16}	4.0	0.979

T5.1

5.2 連続性

以下で扱う浮動小数点数は，符号部，指数部，仮数部の順に並んでおり，指数はゲタはかせで，負の数はその絶対値を表わす浮動小数点表示を32ビットの整数と見て2の補数を取ったもので表わす．指数部のビット数は2進法では9，4進法では8，16進法では7で，ゲタは $(FF)_{16}$，$(7F)_{16}$，$(3F)_{16}$ であるとする．

グラフのX軸に32ビットの浮動小数点数を，符号ビットの直後に小数点がある固定小数点数と見た値を取り，Y軸にはその浮動小数点数の値を取る．このグラフで示される関数を fg とする．このグラフと精度の関係は，グラフの傾きを t とするとXが 2^{-31} だけ変わるとYは $t \times 2^{-31}$ 変化するので，最大絶対誤差は $t \times 2^{-32}$ になる．今まで述べてきた各方式の



G 5

グラフをG5に示す．　正規化したときには現われない部分は点線にした．

これらのグラフから以下のことがわかる．

- Sg_bのグラフはSn_bのグラフの頂点を結んだものを$b/(b-1)$倍したものである．
- Sg_bのグラフが表わす関数fgは$(-1,0)$，$(0,1)$で連続で単調増加である．　但しxは離散的なので$fg(x)$も離散的な値しか取れない．
- Sg_bは「丸め」や「大小の比較」を整数と思って行なえばよい．
- Sg_bでは2つの異なるビットパターンが異なる実数に対応する．
- $Sn_2 \leqq Sn_4 \leqq Sg_4$，$Sn_b \leqq Sg_b$である．　ここで$Sa \leqq Sb$

とは S_b の精度が $[-f_{max}, -f_{min}]$, $[f_{min}, f_{max}]$ の範囲内で常に S_a より良いか，等しいことを示す．

§6 まとめ

以上の結果から下記のことがわかる．

- 実用性を加味して考えると2進けち方式 (S_{g2})，4進方式 (S_{n4}) が良く，次は2進方式 (S_{n2})，16進方式 (S_{n16}) であり，16進切り捨て方式は良くない．
- 丸めは形を変えずに利用でき精度を2倍以上にするので36ビット以下ぐらいでは使うべきである．
- 2進けち方式 (S_{g2}) と16進切り捨て方式 (S_{t16}) とでは，だいたい5倍以上精度が違う。
- 2進けち方式 (S_{g2}) 演算を実現する手間は2進方式 (S_{n2}) とあまり変わらない．
- b進けち方式 (S_{gb} 但し $b \neq 2$ とする) では整数が正確に表わせないから良くない．
- b進けち方式 (S_{gb} 但し $b \neq 2$ とする) では仮数と指数から浮動小数点数に変換するのが少し複雑である．

謝辞

いろいろと指導して下さった木村泉助教授と辻尚史助手，およびプログラミングを一部手伝っていただいた角田博保氏に感謝の意を表わしたい．

参考文献

[1] pdp 11/45 processor handbook,p.165.

[2] R.P.Brent,"On the Precision Attainable with Various Floating-Point Number Systems",IEEE Trans. Comput. Vol. C-22,pp.601-607,Jun.1974.

[3] D.E.Knuth,The Art of Computer Programming,Vol.2. Reading,Mass.:Addison-Wesley,1969,pp.30-31.

[4] Microprogram 21MX Computers operating and reference manual,Hewlett-Packard Comp.,Aug.1974.